



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/582,427	06/08/2006	Xiaofeng Guo	42P22635	4846
45209	7590	02/01/2011	EXAMINER	
MISSION/BSTZ			WEI, ZHENG	
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP			ART UNIT	
1279 OAKMEAD PARKWAY			PAPER NUMBER	
SUNNYVALE, CA 94085-4040			2192	
			MAIL DATE	
			DELIVERY MODE	
			02/01/2011	
			PAPER	

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

### Office Action Summary

**Application No.**

10/582,427

**Applicant(s)**

GUO ET AL

**Examiner**

ZHENG WEI

**Art Unit**

2192

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 11/18/2010.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-28 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-28 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

### **DETAILED ACTION**

#### ***Remarks***

1. This office action is in response to the amendment filed on 11/18/2010.
2. Claims 1, 3, 5, 6, 8, 10, 12, 17, 19, 22, 24 and 26 have been amended.
3. Objection to claims 3-7 is withdrawn in view of Applicants' amendment.
4. The 35 U.S.C. 112 second paragraph rejection to claims 3-7, 10-14, 17-21 and 24-28 is withdrawn in view of the Applicant's amendment.
5. 35 U.S.C. § 101 rejection to claims 22-28 is withdrawn in view of Applicants' amendment.
6. Claims 1-28 remain pending and have been examined.

#### ***Response to Arguments***

7. Applicant's arguments filed on 01/18/2010, in particular on pages 8-13, have been fully considered but they are not persuasive. For example:
  - At page 8-9, Applicants submit that the program transformer as recited in claim 8 is not limited to only software implementation and even if the program transformer is implemented by software, it does not mean that it can be interpreted as software program listing per se. However, Examiner's position is that the claim languages have to be given claims their broadest reasonable interpretation. The broadest reasonable interpretation of the claim 8 covers a selection of using pure software implementation (see for example, specification, paragraph [0022], "All or part of an embodiment of the invention

may be **implemented by hardware, software, or firmware, or any combination thereof.**" [emphasis added]) and thus is treated as software program listings per se. Moreover, the claim language merely recites "associate blocks and "sink the wait instruction", but does not explicitly define the blocks or instructions are in the computer memory or perform operation based on the blocks/instructions in the memory. Therefore, such program transformer module including its components color assoicator and code mover can be read as software code and are not statutory.

- At page 12, second paragraph, Applicants submit that "First, Chang merely discloses a large superblock from each trace of basic blocks formed by code duplication (Chang page 268, section 2.4 "Code Scheduling Algorithm"), not a critical section. A superblock is merely formed by duplicating code from the traces of the basic blocks. In contrast, a critical section is a section that is protected to ensure mutual exclusiveness. Examiner would like to thank Applicants for providing detail information for the critical section. However, it is noted that although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). The critical section as recited in the claim can be reasonable interpreted as code block.
- At page 12, third paragraph, Applicants submit that "Second, Chang merely discloses moving code downward across branch operations within a superblock (Chang, page 268, section 2.5 'Code Scheduling Models', first

paragraph), not sinking the wait instruction down the blocks globally to the end of the critical section. Even if the superblock is a critical section, moving the code downward across branch operations within a superblock does not sink the wait instruction down the blocks globally, or across the blocks, to the end of the critical section. Chang merely discloses moving an instruction downward within a superblock, not globally (or across the blocks), and not to the end of the critical section". However, Examiner's position is that Chang discloses moving downward/sinking the operation/instruction by using a dependent constraint (see for example, Chang, page 268, section 2.5 'Code Scheduling Models', first paragraph). That is to say if there is no dependency conditions, the instruction can be move/sink to globally across the all operation/blocks to the end of superblock/critical section as Chang's example of code scheduling model indicated.

- At page 12, forth paragraph, Applicants submit that "Third, Shpeisman merely discloses a colored conflict graph 700 (Shpeisman, paragraph [0034], lines 1-7), not associating blocks of instructions with color information. Shpeisman explicitly discloses that the color corresponds to a hardware resource (Shpeisman, paragraph [0058], lines 1-3), and the assignment of a color to a node represents the assignment of a hardware resource to the data represented by the node (Shpeisman, paragraph [0058], lines 5-8). In contrast, the rejected claims recite "associating blocks of instructions., with color information," indicating that the color a correspond to the blocks of instructions,

not hardware resources such as a data bank (Shpeisman, paragraph [0058], lines 1-3)". However, Examiner's position is that Shpeisman discloses using coloring information (colored conflict graph)(see for example, paragraph [0034-0036]) to create a data layout by a compiler 202 to reduce or eliminate conflicts (see for example, paragraph [0034]"). It can be seen that Shpeisman's data layout (800) is generated during the compilation time by a compiler based on color information (the colored conflict graph (700)). It is obvious that such conflict graph applied to the data layout/code that also can be used to code/instruction in order to reduce conflict as suggest by Shpeisman (see for example, paragraph [0034-0036]).

- At page 12, fifth paragraph, Applicants submit that "Fourth, Midkiff merely discloses a wait instruction to wait until an event occurs (Midkiff, page 1487, right column, Section IV.C "Two synchronization Instruction Sets"), not a wait instruction associated with a memory access". However, Midkiff also discloses the wait instruction is place before the set instruction and such set instruction is a memory access instruction to set bit (see for example, p.1489, left column, S1, S2 and related text). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to understand that wait instruction is associated with the memory access (see for example, p.1487, "...waits until the bit for the iteration executing the dependence source has been set").

***Claim Rejections - 35 USC § 101***

8. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

9. Claims 8-14 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

**Claim 8:**

Claim 8 recites a program transformer which comprises a color associator, a compensator and a code mover (see for example, paragraph [0031], "The main memory 30 may include...It may include a compiler or program translator or program transformer 135 to compile, translate, or transform the program code..."; also see paragraph [0022], "All or part of an embodiment of the invention may be implemented by hardware, software, or firmware, or any combination thereof."). Such components are only software modules and thus can be interpreted as software program listings per se. The pieces of software are not physical "things". They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer software and hardware components which permit the computer program's functionality to be realized. Therefore, computer programs (transformer) claimed as computer listings per se are nonstatutory. See M.P.E.P. 2106.01 (I)

**Claims 9-14:**

Claims 9-14 depend on claim 6. These claims all fail to remedy the 35 USC 101 nonstatutory problem of claim 8. Therefore, they are also rejected for the same reason.

*--These rejections can be overcome by claiming the program transformer as a computer program product which stores on a non-transitory computer readable storage medium or claiming as a computer system which contains program transformer and computer hardware to execute and realize its functions.*

### ***Claim Rejections - 35 USC § 103***

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

11. Claims 1-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chang (Chang et al., IMPACT: An Architectural Framework for Multiple-Instruction-Issue Processors") in view of Shpeisman (Shpeisman et al., US 2005/0149916 A1) and further in view of Midkiff (Midkiff et al., Compiler Algorithms for synchronization)

Claim 1:

Chang discloses a code scheduling method comprising:

- Forming a program trace from blocks of instructions (basic blocks) which are between start and end of a critical section (superblock) (see for example, p.268, right column, lines 4-8, "Both prepass and postpass code scheduling



algorithms consist of the following steps: 1) Form traces from basic blocks...2) Form a large superblock from each trace of basic blocks..."); and

- sinking (downward code motion) the instruction down the blocks globally across the blocks to the end of the critical section using a dependence constraint on the instruction. (see for example, p.268, section 2.5, lines 1-13, "Our code scheduler moves code both upward and down ward across branch operations within a superblock...For downward code motion, e.g., X precedes Y, if Y does not depend on X then X can be moved below Y..."),

but Chang does not explicitly disclose associating blocks of instruction with color information and the blocks contain a wait instruction associated with a memory access. However, Shpeisman in the same analogous art discloses using coloring information (colored conflict graph)(see for example, paragraph [0034-0036]) to create a data layout by a compiler 202 to reduce or eliminate conflicts (see for example, paragraph [0034]). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use the color information to sink (move) the instruction. One would have been motivated to do so to perform code scheduling without conflicts as suggested by Shpeisman (see for example, paragraph [0034]). Neither Chang nor Shpeisman discloses a wait instruction. However, Midkiff in the same analogous art discloses a wait instruction for compiling program code (see for example, p.1488, section B., Generation of wait and set instructions). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to including

the wait instruction in Chang's code scheduling for code compilation. One would have been motivated to do so to synchronize between blocks of instructions (groups of statement) as suggested by Midkiff (see for example, p.1487, section C., Two Synchronization Instruction sets, lines 3-4)

Claim 2:

Chang discloses the method of claim 1 wherein associating the blocks comprises:

- identifying a sequence of the blocks corresponding to the program trace from a starting block at the start of the critical section to an ending block at the end of the critical section, the starting block containing the wait instruction (see for example, p.268, right column, lines 4-9, "Form traces from basic blocks that are likely to be executed as a sequence...Basic blocks within a superblock are placed sequentially in memory"); and
- constructing a dependency graph for each superblock (see for example, (see for example, p.268, right column, lines 9-11, "3) Construct a dependence graph for each superblock"),

but Chang does not explicitly disclose assigning a color to the sequence of the blocks and the wait instruction. However, as discussed above Shpeisman in the same analogous art discloses assigning color for each of the node in the graph. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to assign color information to each superblock

in Chang' dependency graph. One would have been motivated to do so to perform code/instruction scheduling including wait instruction without conflicts as suggested by Shpeisman (see for example, paragraph [0034]).

Claim 3:

Chang discloses the method of claim 1 wherein sinking the wait instruction comprises:

- speculatively moving the wait instruction to a basic block having multiple predecessor blocks, the multiple predecessor blocks including the starting block (see for example, p.268, section 2.5, lines 1-13, "Our code scheduler moves code both upward and down ward across branch operations within a superblock...For downward code motion, e.g., X precedes Y, if Y does not depend on X then X can be moved below Y...");
- inserting compensation code to at least one of the multiple predecessor blocks excluding the starting block (see for example, p.268, section 2.5, lines 9-13, "Note that if X is to be scheduled after Y and the destination register of X is in live-out(Y), a copy of X must be inserted between Y and its target instruction"); and
- improving the dependence graph by removing dependence and compute live-variable in formation (see for example, p.268, right column lines 10-14), but Chang does not explicitly disclose updating the color information. However, it would have been obvious to one having ordinary skill in the art at the time the

invention was made to update improved dependence graph by using Shpeisman's method. One would have been motivated to do so to perform code/instruction scheduling including wait instruction without conflicts as suggested by Shpeisman (see for example, paragraph [0034]).

Claim 4:

Chang discloses the method of claim 3 wherein speculatively moving the wait instruction comprises:

- moving the wait instruction to the basic block if the starting block and the wait instruction have same color and if the wait instruction is ready (see for example, p.268, section 2.5, lines 1-13, "For downward code motion, e.g., X precedes Y, if Y does not depend on X then X can be moved below Y...") .

Claim 5:

Midkiff discloses the method of claim 3 wherein inserting the compensation code comprises:

- inserting a send signal to the at least one of the multiple predecessor blocks excluding the starting block (see for example, p.1487, section V. Inserting synchronization instructions; also see p.1487, right column section 2), "The wait and set instructions: The set instruction is used to signal that some event has occurred...").

Claim 6:

Shpeisman discloses the method of claim 3 wherein updating the color information comprises:

- resetting the color of the basic block (see for example, Fig.12, step 1202, Coloring the conflict graph to generate a colored conflict graph by assigning a color to each of the plurality of node, each color representing a hardware resource"); and
- resetting the color of the wait instruction having an associated memory access instruction in the basic block (see for example, Fig.12, step 1202, Coloring the conflict graph to generate a colored conflict graph by assigning a color to each of the plurality of node, each color representing a hardware resource").

Claim 7:

Shpeisman discloses the method of claim 6 wherein updating the color information. Change discloses the dependence graph represent a superblock (instructions) (see for example, p.268, right column lines 10-12). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to understand that changing the color of a node by using Shpeisman's method can also be used to change the color of Change's superblock (instructions) including the wait instruction.

Claims 8-14:

Claims 8-14 are computer program transformer component of the claimed method, wherein all claimed limitation functions have been addressed in claims 1-7 above respectively. It is well known in the computer art that such method steps can be implemented as computer program transformer and can be practiced and /or stored on a machine-accessible medium. Thus, they also would have been obvious in view of reference teachings above.

Claims 15-21:

Claims 15-21 are system version for performing the claimed method as in claims 1-7 addressed above, wherein all claimed limitation functions have been addressed and/or set forth above and certainly a computer system would need to run and/or practice such function steps disclosed by reference above. Thus, they also would have been obvious.

Claims 22-28:

Claims 22-28 are an article of manufacture for performing the claimed method as in claims 1-7 addressed above, wherein all claimed limitation functions have been addressed and/or set forth above and certainly an article of manufacture including a machine-accessible medium would need to run and/or practice such function steps disclosed by reference above. Thus, they also would have been obvious.

***Conclusion***

12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.
13. Applicant's arguments with respect to claims rejection have been fully considered but they are not persuasive. Applicant's amendment necessitated additional clarification and/or the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a). A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.
14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Zheng Wei whose telephone number is (571) 270-1059 and Fax number is (571) 270-2059. The examiner can normally be reached on Monday-Thursday 8:00-15:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571- 272-1000.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Z. W./  
Examiner, Art Unit 2192

/Tuan Q. Dam/  
Supervisory Patent Examiner, Art Unit 2192